

### Amendments to the Specification

Please replace paragraph beginning on line 7 of page 12 with the following amended paragraph:

However, if it is determined at step 54 that the code cache is not full, the program analysis with data caching system 50 then translates the code and puts the translated code into the code cache [22] 27, at step 56, as herein defined above. After translating the original source code 22 and placing the translated code into the code cache 27, the program analysis with data caching system 50 then proceeds to step 57.

Please replace paragraph beginning on line 12 of page 12 with the following amended paragraph:

At step 57 the program analysis with data caching system 50 checks whether backpatching the "B" pointer is necessary at step 57. Backpatching [in both] is the process of modifying the destination of the branch address to a different location so that a future lookup of the translation for the destination is avoided. Determination of backpatching is herein defined in further detail with regard to Fig. 6.

Please replace paragraph beginning on line 17 of page 14 with the following amended paragraph:

Illustrated in FIG. 6 is a flow chart of an example of the process that determines if a target address within the code cache has been established and some branches can be backpatched (i.e., the "B" pointer if necessary), as shown in FIG 4. The program analysis with data caching system 50 first tests if the backpatch pointer "B" is equal to null at step 71. If the backpatch pointer "B" is equal to null, the program analysis with data caching system 50 continues the execution of the translated source code by returning to step [82] 61 (FIG. 4) from step 99. If the backpatch pointer "B" is not equal to null, then the program analysis with data caching system 50 backpatches the target address of branch "B" to be the address pointed to by the new program counter 26. The backpatch testing process then returns to step [82] 61 in FIG. 4.